

Chapter 28: Information Storage Format in Database and Configuration Files

This chapter introduces the files **UPS/UPD** uses for database and product management. It also describes the format of the information storage in these files, which is in the form of **KEYWORD=VALUE** pairs. The supported keywords are listed and described.

Most of the time, product installers and **UPS** database managers can get all the information they need about a product or about the contents of a database via the `ups list [-K <keywordList>]` command output (described in section 23.11 *ups list*), which is fairly easy to interpret. However, it's helpful to understand the database files when dealing with complex situations. The keywords described in this chapter which appear in the database files also appear in the `ups list -l` output.

28.1 Overview of File Types

The information that **UPS** needs in order to configure and manage a database and to identify, locate, and retrieve product instances resides in a set of ASCII files in the **UPS** database. The information that **UPD** needs for installing products also resides there. The files used for these purposes include:

- *Version files* tell **UPS** where to find all the files associated with a particular version of a product on the local system, and contain some other information specific to the local installation of the product. They are generally named according to the scheme `vx_y.version`, e.g., `v1_0.version`. These are described in Chapter 29: *Version Files*.
- *Chain files* are optional and contain pointers to version files, thus providing convenient access to particular product versions on the local system. They are generally named according to the scheme `chainname.chain`, e.g., `current.chain`. These are described in Chapter 30: *Chain Files*.
- The *UPS database configuration file* defines things such as which nodes can access products maintained in the database, and which directories house products, **man** pages, **UPS** initialization files, the **UPD** configuration file, and so on. It is described in Chapter 31: *The UPS Configuration File*.

- The *UPD configuration file* controls where **UPD** installs products and miscellaneous product-related files. It can also be used to define supplementary actions for **UPD** to perform when installing or updating products. It is described in Chapter 32: *The UPD Configuration File*.

These files are sometimes referred to collectively as *UPS database files*. They store information in the format of *keywords*.



This information storage format is also used in *table files*, which are provided by the product developer and discussed in Part VIII *Developer's Reference*. They contain product-specific, system-independent information. Table files can be maintained in the database, but they are not constrained to reside there, and in fact usually reside under the product root directory.

28.2 Keywords: Information Storage Format

28.2.1 What is a Keyword?

UPS/UPD utilizes a set of keywords that collectively store the information **UPS/UPD** requires for managing products. A *keyword* represents a category of information used by **UPS/UPD**, it is akin to a variable.¹ A *keyword line* in a file assigns a value to a keyword in the format `KEYWORD = VALUE`.

The supported keywords are listed and described in the table in section 28.4 *List of Supported Keywords*. Some of the keywords can be used in all the file types, others are restricted to certain file types. A few keywords have default values.



Keywords and their values are **not** case-sensitive.

28.2.2 Keyword Syntax

When two or more words are used to make up one keyword, they are generally separated by an underscore (`_`) for readability. All the provided keywords use full words except:

DB	is used instead of DATABASE
DIR	is used instead of DIRECTORY
PROD	is used instead of PRODUCT

1. And in many cases a keyword has an associated read-only variable usable in functions in the table file and/or the `updconfig` file.

28.2.3 User-Defined Keywords

In addition to those listed, **UPS/UPD** allows user-defined keywords (where *user* in this context refers to a product developer or administrative user). All user-defined keywords must have underscore (`_`) as the initial character. While parsing, any unrecognized (i.e., user-defined) keywords are ignored by **UPS**, but they are preserved across rewrites of the files.

28.2.4 How UPS/UPD Sets Keyword Values

Keywords stored in the **UPS** database configuration file (described in Chapter 31: *The UPS Configuration File*) and the **UPD** configuration file (described in Chapter 32: *The UPD Configuration File*) are given values according to the configuration chosen when **UPS/UPD** was installed and configured. See Chapter 14: *Installing UPS and UPD from Bootstrap* for information on choosing values during the installation of **UPS/UPD**.

Keywords stored in version or chain files are set at the time that the corresponding product instance and/or chain is declared to the **UPS** database. Those stored in table files are usually set by the product developer. If a keyword is stored in both the database configuration file and another file, then, for the corresponding product instance(s), the value set at product or chain declaration overrides the one set in the database configuration file.

28.3 Flexibility of File Syntax

The syntax of the database files is *fixed* but *forgiving*. It is *fixed* in the sense that **UPS** commands automatically create the version and chain files in a particular **UPS**-supported format. Any **UPS** command that modifies information in these files rewrites the file to disk according to the same format. The syntax is *forgiving*, however, in that when you perform manual file updates, **UPS** will ignore blank lines and extra whitespace (spaces and tabs). Comment lines can be placed anywhere in the file and must begin with a pound sign (`#`). However, if you want comments to be preserved upon rewrite, they must be the first lines in the file.

28.4 List of Supported Keywords

The following table gives information about each provided keyword. The last five columns indicate which database file the keyword may be used in. The headings D, U, C, V and T refer to:

D	Database configuration file (<code>dbconfig</code>)
U	UPD configuration file (<code>updconfig</code>)
C	Chain file
V	Version file
T	Table file

Keyword and Default Value (if any)	Description and Notes (if any)	D	U	C	V	T
ACTION	defines an action (described in Chapter 34: <i>Actions and ACTION Keyword Values</i>), i.e., groups together a list of functions associated with a command (e.g., <code>ACTION=SETUP</code>)		U			T
ARCHIVE_FILE	archive file name/location; used by UPD				V	
AUTHORIZED_NODES Default: All nodes (*); taken from UPS database configuration file	authorized nodes	D			V	
CATMAN_SOURCE_DIR Default: under the <code>#{UPS_UPS_DIR}/toman/catman</code> directory	location of catman files (formatted man page files) included with instance					T
CATMAN_TARGET_DIR	directory into which catman files are to be copied	D				
CHAIN	chain name			C		
COMMON:	groups together actions that apply to all instances represented in “GROUP:”; COMMON: is only valid within a GROUP:		U			T
COMPILE_DIR	directory in which the compile file resides				V	
COMPILE_FILE	the name of the file containing compiled functions (see Chapter 38: <i>Use of Compile Scripts in Table Files</i>)				V	

Keyword and Default Value (if any)	Description and Notes (if any)	D	U	C	V	T
DECLARED Default: current date and time	the date/time that the instance was declared to UPS or declared with a chain Note: often has multiple values, one for each declaration (e.g., for subsequent chain declarations)			C	V	
DECLARER Default: current user	userid of user that performed the declaration Note: often has multiple values, one for each declaration (e.g., for subsequent chain declarations)			C	V	
DESCRIPTION	product description		U	C	V	T
END:	marks the end of a “GROUP:” or “COMMON:”; one “END:” marker is used to jointly end a “GROUP:” and an included “COMMON:”		U			T
FILE	type of file (possible values: DBCONFIG, UPDCONFIG, CHAIN, VERSION, TABLE)	D	U	C	V	T
FLAVOR	product instance flavor Note: To easily accommodate flavor-neutral setup functions in a table file, FLAVOR can take the value ANY, but <i>only</i> in a table file.		U	C	V	T
GROUP:	groups together multiple instances; all entries subsequent to this “GROUP:” are part of it until an “END:” marker is reached		U			T
HTML_SOURCE_DIR Default: under the $\${UPS_UPS_DIR}$ / tohtml directory	location of html files included with instance <i>not supported in UPS v4</i>					T
HTML_TARGET_DIR	directory into which html files are to be copied <i>not supported in UPS v4</i>	D				
INFO_SOURCE_DIR Default: under the $\${UPS_UPS_DIR}$ / toInfo directory	location of Info files included with instance					T
INFO_TARGET_DIR	directory into which Info files are to be copied	D				

Keyword and Default Value (if any)	Description and Notes (if any)	D	U	C	V	T
MAN_SOURCE_DIR Default: under the \${UPS_UPS_DIR} / toman/man directory	location of unformatted man page files included with instance					T
MAN_TARGET_DIR	directory into which formatted man pages are to be copied	D				
MODIFIED Default: Current date/time	last time the associated instance was changed Note: often has multiple values, one for each declaration/modification (e.g., for subsequent chain declarations)			C	V	
MODIFIER Default: Current user	userid of user that modified the instance Note: often has multiple values, one for each declaration/modification (e.g., for subsequent chain declarations)			C	V	
NEWS_SOURCE_DIR Default: under the \${UPS_UPS_DIR} / tonews directory	location of news files included with instance <i>not supported in UPS v4</i>					T
NEWS_TARGET_DIR	directory into which news files are to be copied (for posting to a newsgroup) <i>not supported in UPS v4</i>	D				
ORIGIN	master source file; see option -D in Chapter 25: <i>Generic Command Option Descriptions</i>				V	
PRODUCT	product name		U	C	V	T
PROD_DIR	product root directory (usually defined relative to PROD_DIR_PREFIX, below)				V	
PROD_DIR_PREFIX	product root directory prefix (area where all or most product instances are maintained)	D				
QUALIFIERS	additional instance specification information often used to indicate compilation options used by developer Notes: appears immediately after a FLAVOR in these files, and is coupled with it to complete the instance identification (see 27.2.3 <i>Qualifiers: Use in Instance Matching</i>)		U	C	V	T
SETUPS_DIR	location of setups.[c]sh files and other UPS initialization files	D				

Keyword and Default Value (if any)	Description and Notes (if any)	D	U	C	V	T
STATISTICS	flag to record statistics for specified products See 28.6.3 <i>STATISTICS</i> for usage information.	D			V	
TABLE_DIR Default: search path (see section 29.4 <i>Determination of ups Directory and Table File Locations</i>)	location of table file				V	
TABLE_FILE	name of table file (relative to TABLE_DIR)				V	
UNWIND_ARCHIVE_FILE	(a UPD keyword used only on distribution server configurations) absolute path to directory in which to unwind archive file (tar file) of product		U			
UNWIND_PROD_DIR	(a UPD keyword) absolute path to directory where product gets unwound		U			
UNWIND_TABLE_DIR	(a UPD keyword) absolute path to directory where the table file gets unwound		U			
UNWIND_UPS_DIR	(a UPD keyword) absolute path to directory where the <code>ups</code> directory gets unwound		U			
UPD_USERCODE_DB	Database containing UPD_USERCODE_DIR (set internally)					
UPD_USERCODE_DIR	Directory where UPD configuration files are maintained	D				
UPS_ARCHIVE_FILE	(a UPD keyword used only on distribution server configurations) archive file (tar file) location that UPD specifies in ups declare -T ftp://host\${UPS_ARCHIVE_FILE}		U			
UPS_DB_VERSION	UPS database version	D		C	V	T
UPS_DIR Default: \${UPS_PROD_DIR} /ups if directory exists there	location of <code>ups</code> directory (if not absolute path, then taken relative to PROD_DIR, if specified)				V	

Keyword and Default Value (if any)	Description and Notes (if any)	D	U	C	V	T
UPS_PROD_DIR	(a UPD keyword) product root directory that UPD specifies in the ups declare -r option; should be defined relative to PROD_DIR_PREFIX for portability		U			
UPS_TABLE_DIR	(a UPD keyword) table file directory that UPD specifies in the ups declare -M option Normally this should not be set! Since UPS_TABLE_DIR must be an absolute path, the declaration becomes non-portable if you set this location.		U			
UPS_THIS_DB	(a UPD keyword) the database into which UPS declares the product (i.e., the directory that UPD specifies in the ups declare -z option).		U			
UPS_UPS_DIR	(a UPD keyword) ups directory that UPD specifies in the ups declare -U option, taken relative to $\${UNWIND_PROD_DIR}$ unless an absolute path is given; usually defined as ups .		U			
UPS_TABLE_FILE	(a UPD keyword) table file name that UPD specifies in the ups declare -m option		U			
USER	current username					T
VERSION	product version			C	V	T
_UPD_OVERLAY	main product name for overlaid product Note: This keyword is user-defined from UPS 's point of view. It is included here because it is configured and used by UPD . Its use with overlaid products is described in section 28.6.6 <i>_UPD_OVERLAY</i> .					T

28.5 Syntax for Assigning Keyword Values

- Any keyword value that has multiple values uses a colon (:) to separate the subvalues. The value (i.e., the list of subvalues) may be surrounded by double quotation marks (" . . . "). Blanks within the double-quoted value are ignored; they are neither required nor prohibited.

For example, the following are all equivalent:

```
QUALIFIERS = debug:optimize
```

```
QUALIFIERS = "debug:optimize"
```

```
QUALIFIERS = " debug: optimize"
```

- Whitespace is ignored except within the keyword values for DESCRIPTION, DECLARER and MODIFIER
- Leading whitespace is ignored.
- There are no line continuation characters; the entire keyword definition or function must appear on a single line.
- The “at” character (@) is defined for use with the keywords COMPILE_FILE, PROD_DIR, UPS_DIR and TABLE_FILE. See section 28.6 *Usage Notes on Particular Keywords*.



28.6 Usage Notes on Particular Keywords

28.6.1 COMPILE_DIR, COMPILE_FILE and @COMPILE_FILE

COMPILE_DIR	the directory in which the compile file resides (see Chapter 38: <i>Use of Compile Scripts in Table Files</i>)
COMPILE_FILE	the name of the file containing precompiled functions
@COMPILE_FILE	the entire path to the file containing precompiled functions

28.6.2 PROD_DIR_PREFIX, PROD_DIR and @PROD_DIR

PROD_DIR_PREFIX is generally set to the root of the path shared by all the products.

PROD_DIR is the path that gets specified when the particular product instance is declared; it is usually (but not always) a relative path that gets tacked onto PROD_DIR_PREFIX.

@PROD_DIR is a shorthand to request the entire path for the directory where the product is installed (usually equivalent to PROD_DIR_PREFIX/PROD_DIR).



If `PROD_DIR_PREFIX` is not defined on your system, then `PROD_DIR` should represent the entire path, in which case `PROD_DIR` and `@PROD_DIR` are identical.

Products installed prior to the upgrade to **UPS** v4 often reside in a different area than the newer products, and you may find that `PROD_DIR_PREFIX` is not set properly for them.

Compare these commands and their output:

```
% ups list -K PROD_DIR_PREFIX teledata
    "/afs/fnal.gov/ups/prd"
% ups list -K PROD_DIR teledata
    "teledata/v1_0/NULL"
% ups list -K @PROD_DIR teledata
    "/afs/fnal.gov/ups/prd/teledata/v1_0/NULL"
```

28.6.3 STATISTICS

The `STATISTICS` keyword is provided to allow recording of the following statistics on product usage and **UPS** database access:

- Userid of person executing **UPS/UPD** command
- Date and time

- Which command was executed (including options and arguments)
- Which product instance was selected by command

This keyword can appear in a product's version file and/or in the **UPS** database configuration file, thus providing a great deal of flexibility in choosing which products/instances to monitor.

Use in a Version File

When the **STATISTICS** keyword is present in a version file, it must be included with each specific instance which is to be monitored. If the **STATISTICS** keyword is located *before* any **FLAVOR** and/or **QUALIFIERS** keywords (these keywords separate out different instances), then it is ignored. In a version file, this keyword should have no value assigned.

Use in a Database Configuration File

When the **STATISTICS** keyword appears in the database configuration file, it needs a value. (If it has no value, it is ignored.) Its value is a colon-separated list of the products (name only) on which to record statistics (e.g., `STATISTICS = "tcl:tk:cern"`). The value `*` (asterisk) indicates that statistics are to be gathered on all products in the database.

Statistics Output

For a given product being monitored, statistics data for the product get recorded in a file whose name is the same as the product. If the product has dependencies, data also get recorded for them in their own product-specific files, and the data include the parent product name and version number. The data get recorded only when the **UPS/UPD** command in question has succeeded (i.e., when the temporary file has been created, but not yet sourced).

The statistics output files for all the monitored products and their dependencies reside in a special directory associated with the **UPS** database, namely `$PRODUCTS/.upsfiles/statistics`. This makes it easy to determine which products are being monitored, and only one directory needs to be made world-writable.

As an example of the statistics data that get recorded, let's look at the **tcl** product. It is a dependency of **tk**. Data that are recorded when an instance of **tcl** is accessed independently look like this:

```
"tcl" "v8_0" "IRIX" "" "" "user1" "2000-03-18 15.22.36 GMT"
"setup"
```

Data that are recorded for **tcl** when an instance of **tk** is accessed look like this:

```
"tcl" "v8_0" "IRIX" "" "" "user1" "2000-03-18 15.22.36 GMT"
"setupRequired tk v8_0"
```

28.6.4 TABLE_FILE and @TABLE_FILE

TABLE_FILE represents only the name of the table file, not its path.
@TABLE_FILE is the entire path for the table file. Compare these commands and their output:

```
% ups list -Ktable_file teledata
    "v1_0.table"
% ups list -K@table_file teledata
    "/afs/fnal.gov/ups/db/teledata/v1_0.table"
```

See section 29.4 *Determination of ups Directory and Table File Locations* for information on how UPS determines the table file directory.

28.6.5 UPS_DIR and @UPS_DIR

UPS_DIR represents the location of the product's ups directory. If it is not an absolute path, then it is taken relative to @PROD_DIR (as shown in the example below). @UPS_DIR is the absolute path. Compare these commands and their output:

```
% ups list -K @PROD_DIR teledata
    "/afs/fnal.gov/ups/prd/teledata/v1_0/NULL"
% ups list -Kups_dir teledata
    "ups"
% ups list -K@ups_dir teledata
    "/afs/fnal.gov/ups/prd/teledata/v1_0/NULL/ups"
```

28.6.6 _UPD_OVERLAY

The _UPD_OVERLAY keyword defined in **UPD**¹ is provided for inclusion in the table file of each overlaid product. Overlaid products are introduced in section 2.3.7 *Product Overlays* and discussed again for developers in section 17.2.4 *Overlaid Products*. _UPD_OVERLAY takes as its value the main product name in double quotes. Its presence indicates that the product is an overlaid product maintained in the root directory of the main product listed as the keyword's value. For example, the table files for the products **cern_bin**, **cern_ups**, and **cern_lib** would contain the following keyword line:

```
_UPD_OVERLAY = "cern"
```

UPD would then use **cern** as the product name when determining the root directory.

1. UPS regards the _UPD_OVERLAY keyword as user-defined.

